

(translation of the front page of the priority document of
Japanese Patent Application No. 2000-402713)

PATENT OFFICE
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of the
following application as filed with this Office.

Date of Application: December 28, 2000

Application Number : Patent Application 2000-402713

Applicant(s) : Canon Kabushiki Kaisha

December 21, 2001

Commissioner,
Patent Office

Kouzo OIKAWA

Certification Number 2001-3110652

CFM 2450 VS

日 本 国 特 許 庁

JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年12月28日

出 願 番 号

Application Number:

特願2000-402713

出 願 人

Applicant(s):

キヤノン株式会社

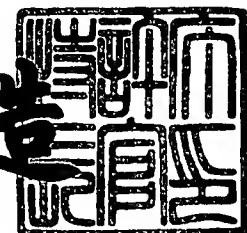


CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年12月21日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



【書類名】 特許願

【整理番号】 4273098

【提出日】 平成12年12月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00

【発明の名称】 ユーザインタフェース制御装置および方法ならびに記憶媒体

【請求項の数】 8

【発明者】

 【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社
社内

 【氏名】 皆川 智徳

【特許出願人】

 【識別番号】 000001007

 【氏名又は名称】 キヤノン株式会社

【代理人】

 【識別番号】 100076428

 【弁理士】

 【氏名又は名称】 大塚 康徳

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100101306

 【弁理士】

 【氏名又は名称】 丸山 幸雄

 【電話番号】 03-5276-3241

【選任した代理人】

 【識別番号】 100115071

 【弁理士】

 【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0001010

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ユーザインタフェース制御装置および方法ならびに記憶媒体

【特許請求の範囲】

【請求項 1】 所定の制御対象に対する設定情報を入力可能なユーザインタフェースを介して、入力された設定情報により生じる不整合を回避するユーザインタフェース制御装置であって、

不整合回避戦略を示すコンフリクト処理ルールを記憶する記憶手段と、

前記入力された設定情報に基づき前記コンフリクト処理ルールを適用することで、関連する設定情報を更新する更新手段と、を有し、

該更新手段は、

前記コンフリクト処理ルールの適用により変更すべき設定情報を検出する検出手段と、

該検出された設定情報のみを変更する設定情報変更手段と、

を有することを特徴とするユーザインタフェース制御装置。

【請求項 2】 前記設定情報変更手段により設定情報が変更されたことを通知する通知手段を更に有することを特徴とする請求項 1 に記載のユーザインタフェース制御装置。

【請求項 3】 前記制御対象は、画像形成装置であることを特徴とする請求項 1 または 2 に記載のユーザインタフェース制御装置。

【請求項 4】 所定の制御対象に対する設定情報を入力可能なユーザインタフェースを介して、入力された設定情報により生じる不整合を回避するユーザインタフェース制御方法であって、

不整合回避戦略を示すコンフリクト処理ルールを記述したコンフリクト処理ルール記述ファイルを参照し、該コンフリクト処理ルールの適用により変更すべき設定情報を検出する検出工程と、

該検出された設定情報のみを変更する設定情報変更工程と、

を有することを特徴とするユーザインタフェース制御方法。

【請求項 5】 前記設定情報変更工程により設定情報が変更されたことを通知する通知工程を更に有することを特徴とする請求項 4 に記載のユーザインタフ

エース制御方法。

【請求項 6】 前記コンフリクト処理ルール記述ファイルは、所定の設定情報の変更を規制する制御コマンドの記述を含むことが可能であって、

前記検出工程は、前記コンフリクト処理ルール記述ファイルより読み出した制御コマンドに応じて該当する設定情報の変更を規制する工程を有することを特徴とする請求項 4 または 5 に記載のユーザインタフェース制御方法。

【請求項 7】 所定の制御対象に対する設定情報を入力可能なユーザインタフェースを介して、入力された設定情報により生じる不整合を回避するユーザインタフェース制御方法をコンピュータによって実現するためのプログラムを格納した記憶媒体であって、

不整合回避戦略を示すコンフリクト処理ルールを記述したコンフリクト処理ルール記述ファイルを参照し、該コンフリクト処理ルールの適用により変更すべき設定情報を検出する検出工程のプログラムコードと、

該検出された設定情報のみを変更する設定情報変更工程のプログラムコードと

を格納することを特徴とする記憶媒体。

【請求項 8】 前記設定情報変更工程により設定情報が変更されたことを通知する通知工程のプログラムコードを更に有することを特徴とする請求項 7 に記載の記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ユーザインタフェースを介して入力される所定の制御対象に対する設定データの間には生じる不整合を回避するユーザインタフェース制御装置および方法ならびに記憶媒体に関する。

【0002】

【従来の技術】

ユーザインタフェース（以下、「UI」ともいう。）を介してユーザから複数の設定値の入力を受け付け、それらの設定値に基づき制御される機器が知られて

いる。ホストコンピュータに接続され、そのホストコンピュータにおいて設定された情報に基づき画像形成処理を行う画像形成システム（プリンタシステム）はその一例である。プリンタシステムのホストコンピュータは一般に、印刷動作を制御するいわゆるプリンタドライバ、およびユーザからの印刷設定等を受け付けるUIを含む印刷処理関連プログラムを有する。

【0003】

印刷処理関連プログラムは、UIを介してユーザから設定値の入力を受け付ける度に、それまでに設定された複数の設定値の中で関連する設定値の値との関係性を評価し、設定値間の不整合（コンフリクト）がないかどうかを判別する。コンフリクトの例としては、記録媒体としてセットされたOHPシートに対して両面印刷を行わせるような設定といったユーザにとって不都合と予想される設定や、プリンタに不可能な動作を行わせるような設定等がある。

【0004】

コンフリクトがあった場合には、それを回避するためのコンフリクト処理を実施する必要がある。

【0005】

従来より、コンフリクト処理が必要となる複数の設定値の条件（コンフリクト処理ルール）を一覧としてまとめてファイル等に保存しておき、このファイルをコンフリクト処理用プログラムに読み込ませることで、コンフリクト処理を行うものがある。コンフリクト処理によっては、データ構造体中の設定値の変更、さらにはUIのコントロール上の設定値の変更、コントロールのグレイアウト（設定項目を薄いグレイで表示することで当該項目については設定不可であることを通知する）処理等を行うものもある。かかる方式は、コンフリクト処理用プログラムが特定の設定値に依存することなく汎用的に利用できるという点で優れている。

【0006】

コンフリクト処理用のプログラムにおいて、コンフリクト処理を統括するコンフリクトマネージャは一般に、コンフリクト処理ルールを汎用的に利用できるようにメインプログラムから独立して高いメンテナンス性を持つように設計されて

おり、それによってコンフリクトマネージャは、メインプログラムからはブラックボックスとして見えることになる。

【0007】

【発明が解決しようとする課題】

しかしながら実際には、メインプログラムでUIの更新処理を行ったり、コンフリクト処理に関与しない特定の設定値の変更に伴いUIの更新処理が必要になる場合もある。従来このような場合、メインプログラムは、該当する項目のみに絞った処理を行うことはできず、ある範囲全体をリフレッシュすることになるため、メインプログラムの処理効率が悪いという問題があった。

【0008】

メインプログラムとコンフリクトマネージャとの仲立ちとなるデータ構造体の差分から判断することもできるが、この方法でも効率が悪い。そのうねコントロールのグレイアウトや表示/非表示の変更もされるような場合には、それらの抽出そのものが困難であったり、さらに効率が悪くなる場合が多いという問題があった。

【0009】

本発明は上記の問題点に鑑みてなされたもので、コンフリクト処理用プログラムにおけるメインプログラムとコンフリクトマネージャとの独立性を保ったまま、コンフリクト処理によって変更された項目の情報のみを両者間で受け渡し可能とすることで、メインプログラムの処理効率を向上させるユーザインタフェース制御装置および方法ならびに記憶媒体を提供することを目的とする。

【0010】

【課題を解決するための手段】

上記目的を達成するため、例えば本発明のユーザインタフェース制御装置は、以下の構成を備える。すなわち、

所定の制御対象に対する設定情報を入力可能なユーザインタフェースを介して、入力された設定情報により生じる不整合を回避するユーザインタフェース制御装置であって、

不整合回避戦略を示すコンフリクト処理ルールを記憶する記憶手段と、

前記入力された設定情報に基づき前記コンフリクト処理ルールを適用することで、関連する設定情報を更新する更新手段と、を有し、

該更新手段は、

前記コンフリクト処理ルールの適用により変更すべき設定情報を検出する検出手段と、

該検出された設定情報のみを変更する設定情報変更手段と、

を有することを特徴とする。

【 0 0 1 1 】

【発明の実施の形態】

(ハードウェア構成)

図 1 は本発明の一実施形態を示す印刷処理システムのブロック構成図である。印刷処理システムは、ホストコンピュータ 3 0 0 0 と、プリンタ 1 5 0 0 より構成される。

【 0 0 1 2 】

ホストコンピュータ 3 0 0 0 において、1 はシステムバス 4 に接続される各デバイスを総括的に制御する CPU、2 は CPU 1 の主メモリ、ワークエリア等として機能する RAM である。3 は各種プログラム、データを格納する ROM であって、各種フォントを記憶するフォント ROM 3 a、ブートプログラムや BIOS 等を記憶するプログラム ROM 3 b、および各種データを記憶するデータ ROM 3 c に区分けして構成されている。

【 0 0 1 3 】

5 はキーボードコントローラ (KBC) で、キーボード 9 や不図示のポインティングデバイス (マウス) からのキー入力を制御する。6 は CRT コントローラ (CRTC) であり、CRT ディスプレイ (CRT) 1 0 の表示を制御する。

【 0 0 1 4 】

外部メモリ 1 1 (ディスクコントローラ (DKC) 7 によりアクセス制御される) は、ハードディスク (HD) やフロッピーディスク (FD) 等であり、図示の如く、オペレーティングシステムプログラム (以下 OS) 2 0 5 をはじめ各種アプリケーション (例えば、図形、イメージ、文字、表等が混在した文書処理を

行う文書処理アプリケーションプログラム) 201、印刷処理関連プログラム204を記憶する他、ユーザファイル、編集ファイル等も記憶する。印刷処理関連プログラム204は、ページ記述言語を用いて記述される印刷データを生成するプログラムであって、同系列の複数のプリンタに対して共通に利用されうる。また、プリンタ制御コマンド生成モジュール(以下「プリンタドライバ」という)2041、プリンタドライバUI制御モジュール2042を含む。

【0015】

8はプリンタコントローラ(PRTC)であり、双方向性インタフェース21を介してプリンタ1500に接続され、プリンタ1500との通信制御処理を行う。

【0016】

外部メモリ11に記憶されたアプリケーションは、RAM2にロードされてCPU1により実行されることになる。また、CPU1は、例えばRAM2へのアウトラインフォントの展開(ラスタライズ)処理を実行し、CRT10上でのWYSIWYG(What you see is What you get)を可能としている。さらに、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウインドウを開き、種々のデータ処理を実行する。ユーザは印刷を実行する際、印刷設定設定画面(プリンタドライバUI制御モジュール2042により制御される)を開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバ2041に対する印刷処理の設定を行うことができる。

【0017】

プリンタ1500において、12はプリンタ1500の全体を制御するCPUである。19はCPU12の主メモリ、ワークエリア等として機能するとともに、出力情報展開領域、環境データ格納領域、NVRAM等に用いられるRAMであり、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるように構成されている。13はROMであり、各種フォントを記憶するフォントROM13a、制御プログラム等を記憶するプログラムROM13b、および各種データを記憶するデータROM13cより構成される。

【0018】

外部メモリ 1 4（メモリコントローラ（MC） 2 0によりアクセスを制御される）は、オプションとして接続されるハードディスク（HD）、フロッピーディスク（FD）、ICカード等であり、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。ハードディスク等の外部メモリ 1 4が接続されていない場合には、ROM 1 3のデータROM 1 3 cに、ホストコンピュータ 3 0 0 0で利用される情報等を記憶することになる。なお、外部メモリ 1 4は 1 個に限らず、複数備えるものであってもよく、例えば、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラム等を格納した外部メモリを複数接続できるように構成されていてもよい。

【 0 0 1 9 】

操作部 1 5 0 1にはユーザからの操作を受け付ける操作パネルが設けられ、その操作パネルには操作のためのスイッチおよびLED表示器等が配されている（図示は省略）。また、図示しないNVRAMを有し、操作パネル 1 5 0 1からのプリンタモード設定情報を記憶するようにしてもよい。

【 0 0 2 0 】

プリンタCPU 1 2は、ROM 1 3のプログラムROM 1 3 bに記憶された制御プログラム等に基づき、印刷部インタフェース 1 6を介してシステムバス 1 5に接続される印刷部（プリンタエンジン） 1 7に出力情報としての画像信号を出力する。また、CPU 1 2は入力部 1 8を介してホストコンピュータ 3 0 0 0との通信処理が可能となっており、プリンタ 1 5 0 0内の情報等をホストコンピュータ 3 0 0 0に通知可能に構成されている。

【 0 0 2 1 】

（ソフトウェア構成）

図 2 は、所定のアプリケーションおよび印刷処理関連プログラムを起動して、ホストコンピュータ 3 0 0 0上のRAM 2にロードされた状態のRAM 2のメモリマップを示している。RAM 2には、図示の如く、BIOS 2 0 6、OS 2 0 5をはじめ、アプリケーション 2 0 1、印刷処理関連プログラム 2 0 4、および関連データ 2 0 3がロードされており、さらに空きメモリ領域 2 0 2も確保されている。これにより、アプリケーション 2 0 1および印刷処理関連プログラム 2

04 は実行可能状態にある。

【0022】

印刷処理関連プログラム204におけるプリンタドライバUI制御モジュール2042は、ユーザによる印刷設定指令に応じてCRT10に印刷設定画面を表示しユーザからの設定を可能にする。

【0023】

図7に、印刷設定画面の表示例を示す。同図において、[Print Style]欄80は、印刷レイアウトを指定する欄であり、ユーザは例えば、片面印刷(1-Sided Printing)801、両面印刷(2-Sided Printing)802、および製本印刷(Booklet Printing)803のうちのいずれかをマウスでクリックすることで指定することができる。

【0024】

[Finishing]欄81は、印刷済み記録媒体の出力順序および仕上げについて指定する欄であり、ユーザは例えば、以下のいずれかから指定可能となっている。

【0025】

[Collate] 811

部単位印刷。Nページの文書をM部印刷する場合に、第1ページ、第2ページ、…、第Nページの順に1枚ずつ出力し、これをM回繰り返す。

【0026】

[Group] 812

ページ単位印刷。Nページの文書をM部印刷する場合に、第1ページをM枚、第2ページをM枚、…、第NページをM枚、の順に出力する。

【0027】

[Staple] 813

ステープル仕上げ。上記[Collate] 811と同様に部単位で出力し、仕上げとして各部ごとにステープラで止める。

【0028】

なお、本明細書では、上記したようなユーザ設定可能項目を「プリンタ機能」または単に「機能」とよぶ。この他にも多くのプリンタ機能を有するが、説明を

簡単にするため省略する。

【0029】

ここで、ユーザにとって不都合と思われる設定の組み合わせ、意味のない設定の組み合わせ、すなわち設定値間の不整合（コンフリクト）は、プリンタドライバUI制御モジュール2042により、以下詳細に説明するコンフリクト処理として回避されるように設計される。例えば、図示においては、印刷レイアウトとして片面印刷（1-Sided Printing）801が指定されているが、この場合には、[Finishing]欄81の[Staple]813はグレイアウト表示され、指定できないようになる。また、図8に示すように、印刷レイアウトとして製本印刷（Booklet Printing）803が指定されたときには、[Finishing]欄81のすべてがグレイアウト表示され、指定できないようにされる。上記した例はごく簡単な例示であって、実際に想定されるコンフリクトはかなりの数にのぼるであろう。以下、コンフリクト処理の詳細を説明する。

【0030】

図3は、実施形態における印刷処理関連プログラム204のプリンタドライバUI制御モジュール2042の概略構成を示している。303は、各モジュール間のデータの受け渡しやデータの更新等を管理してコンフリクト処理を統括するコンフリクトマネージャである。306が、上記した印刷設定画面表示制御を行い、本プリンタドライバUI制御モジュール2042におけるメインプログラムとしてのプリンタドライバUIである。301は、後述する記述形式で記述される不整合回避戦略を示すコンフリクト処理ルールを列記したコンフリクト処理ルール記述ファイルである。302は、コンフリクト処理ルール記述ファイル301をロードして、入力された設定値に対してコンフリクト処理ルールを適用し、各機能の状態を推論する推論エンジン、304は、各プリンタ機能の状態をリスト形式で表示する状態変数リストであり、ユーザからの入力およびコンフリクト処理ルール記述ファイル301の内容に基づき更新されうる。305は、プリンタドライバUI306が提供する画面表示の基になる帳票としての内部構造体であり、状態変数リスト304と連動して各プリンタ機能の状態を所定の形式で表示する。

【0031】

その後、プリンタドライバUI 306を介してユーザからの設定情報を受け取ったコンフリクトマネージャ303は、コンフリクト処理ルール記述ファイル301を参照する。このことは、図示の如くコンフリクト処理ルール記述ファイル301からコンフリクトマネージャ303に向かう矢印で、「R(Read)」として表示されている。参照の結果、設定情報がコンフリクト処理ルールに適合する場合、コンフリクト処理が適用される。そうして、コンフリクトマネージャ303は、状態変数リスト304および内部構造体305を更新してプリンタドライバUI 306に反映させる。この更新作業は、図示の如くコンフリクトマネージャ303は、状態変数リスト304および内部構造体305とは各々双方向矢印で結ばれ、「R/W (Read/Write)」として表示されている。

【0032】

図4は、図3に示した各モジュールで扱われるデータの関連を説明する模式図である。図4において、コンフリクト処理ルール記述ファイル301は、推論エンジン302にインクルード（ロード）されたかたちで参照され、さらに新たなルールが追記される。このコンフリクト処理ルール記述ファイル301は、コンフリクトマネージャ303にも参照され、それを受けて状態変数リスト304が変更されることになる。また、内部構造体305と状態変数リスト304とは先に述べたとおり連動表示されるものであるから、互いにマッピングされる関係にある。そしてこの状態がプリンタドライバUI 306の制御によってユーザに見えるかたちで表現される。

【0033】

内部構造体305は、プリンタ機能名A, B, C に対応するメンバをそれぞれ cA, cB, cC と表現する。

【0034】

(コンフリクト処理ルールの記述形式)

次に、コンフリクト処理ルール記述ファイル301について説明する。

【0035】

各ルールは、論理（ロジック）を用いて数学的に形式化する。述語は、「プリ

ンタ機能名（引数）」の形で記述される。引数としては、ON/OFFの他、数値が入る場合もある（例えば、印刷部数等）。左辺にはプリンタ機能名（引数）を、右辺には左辺が成り立つための論理条件を記述し、記号「←」または「<-」で関係づける。例えば、

A(ON) ← B(ON).

は、「プリンタ機能Bの状態がONのときは、プリンタ機能Aの状態をONとする」という意味のルールになる。

【0036】

また、式中の記号「,」は「かつ（AND）」の意味で用いられる。例えば、「プリンタ機能Bの状態がON、かつ、プリンタ機能Cの状態がOFFのとき、プリンタ機能Aの状態をONとする」というルールは、

A(ON) ← B(ON), C(OFF).

と記述される。

【0037】

さらに、コンフリクト処理ルール記述ファイル301の中には、プリンタドライバUI306を更新するための処理を記述することが可能である。推論エンジン302がその記述を解釈した時点で、コンフリクトマネージャ303の状態変数リスト304を介して、プリンタドライバUI306の更新処理を直接行うようにしても良い。例えば、UI更新の処理として{disable}の記述を追加することにより、該当項目のコントロールをdisableする処理（グレイアウト表示するとともに設定不可の状態にする処理）を指示することができる。

【0038】

図6は、以上の例に従って記述されている。図7に表示されたプリンタ機能を例にとると、図7に示した[Collate] 811に対応する部単位印刷機能、[Group] 812に対応するページ単位印刷機能、および[Staple] 813に対応するステープル仕上げ機能はそれぞれ、Collate()、Group()、およびStaple()で示され、引数はONまたはOFFとなる。また、[PrintStyle] 欄80に対応する印刷レイアウト機能は、Layout()で示され、引数は、1-Sided、2-Sided、Bookletのいずれかである。

【 0 0 3 9 】

図 6 の (1) は、ユーザにより [Collate] 8 1 1 をチェックされたことで Collate (ON) となったときは、Staple(OFF) とするルールを表す。(2) は、Group(ON) となったときは、同じく Staple(OFF) とするルールである。

【 0 0 4 0 】

また、(3) 以降の各行の次の行には、UI 更新の処理として {disable} の記述が追加されている。この記述により、各行のルールを適用後に該当項目のコントロールが disable されることになる。

【 0 0 4 1 】

(プリンタドライバ UI 制御モジュール 2 0 4 2 の処理の内容)

以下、図 5 のフローチャートを用いて、コンフリクト処理を含むプリンタドライバ UI 制御モジュール 2 0 4 2 の処理について、詳しく説明する。

【 0 0 4 2 】

プリンタドライバ UI 制御モジュール 2 0 4 2 の処理は、ユーザがキーボードコントローラ K B C 5 等を用いて、印刷設定画面を開く指示をすることで始まる。ユーザが印刷設定画面を開くよう指示すると、上記したとおり、OS 2 0 5 の管理の下、RAM 2 に印刷処理関連プログラム 2 0 4 がロードされる。

【 0 0 4 3 】

印刷処理関連プログラム 2 0 4 が RAM 2 にロードされると、まず、印刷設定画面を開くための初期化処理として、推論エンジン 3 0 2 は、コンフリクト処理ルール記述ファイル 3 0 1 をコンフリクトマネージャ 3 0 3 を介して、RAM 2 に読み込む (ステップ S 5 0 1)。

【 0 0 4 4 】

続いて、コンフリクトマネージャ 3 0 3 が利用する状態変数リスト 3 0 4 を作成する。(ステップ S 5 0 3)。

【 0 0 4 5 】

コンフリクト処理ルール記述ファイル 3 0 1 に記述されるすべてのプリンタ機能名は、コンフリクトマネージャ 3 0 3 の内部にインクルードされた状態変数リスト 3 0 4 にそれぞれ、状態変数を持っている。この状態変数の値は、プリンタ

ドライバUI 3 0 6で使用される内部構造体3 0 5の対応するメンバの値と連動している。各プリンタ機能名の状態変数の初期値は、その内部構造体3 0 5のメンバの値となる。

【0 0 4 6】

例えば、図4において、内部構造体3 0 5に記述されているint caの初期値は0であるから、それに対応する状態変数リスト3 0 4におけるプリンタ機能Aの値はOFFとなっている。したがって、推論エンジン3 0 2内で保持されるプリンタ機能Aの状態の初期値はOFFとなる。同様に、プリンタ機能名Bの初期値はON、プリンタ機能名Cの初期値はOFFとなる。すなわち、

A OFF
B ON
C OFF

となる。

【0 0 4 7】

その後、推論エンジン3 0 2は、コンフリクト処理ルール記述ファイル3 0 1を参照することでコンフリクトチェックの推論を行う。例えば、図4に示すように、コンフリクト処理ルール記述ファイル3 0 1に記述されている、

A(ON) ← B(ON), C(OFF).

が成立した場合、推論エンジン3 0 2は、状態変数リスト3 0 4中のプリンタ機能Aの状態変数値を、初期値OFFからONに変更する。すなわち、

A ON
B ON
C OFF

となる。このように状態変数リスト3 0 4が初期化される。

【0 0 4 8】

コンフリクトチェックの推論が終了した後、コンフリクトマネージャは、変更された状態変数の値を内部構造体3 0 5の対応するメンバ int caに反映させる。すなわち、int ca は上記ルールが成立したことによって0から1に変更される。

【0049】

続いて、印刷設定画面のオープンのために必要とされるその他の初期化処理を行い、図7に例示したような印刷設定画面をオープンする（ステップS504）。

【0050】

印刷設定画面がオープンされた後は、OSより送られてくるイベントを取得し、そのイベントに対する処理を行う（ステップS505）。

【0051】

次に、ステップS505にて取得したイベントが、ユーザが印刷設定画面上の設定項目を変更したイベントであるか否かの判別を行い（ステップ506）、印刷設定画面上の設定項目を変更するイベントではなかった場合には、ステップS515に進み、印刷設定画面のクローズ要求か否かの判別を行う。クローズ要求であった場合には、ステップS516に進み、終了処理を行い、印刷設定画面をクローズして、すべての処理を終了する。一方、ステップS515での判別の結果、クローズ要求ではなかった場合は、ステップS505に戻り、処理を繰り返す。

【0052】

ステップS506で、ステップS505で取得したイベントがユーザによる設定変更要求であった場合には、ステップS507に進み、構築したコンフリクト処理ルールを適用する。

【0053】

取得したイベントがユーザによる設定変更要求であった場合の一例として、図7に示す[Print Style]欄80における、片面印刷（1-Sided Printing）801から製本印刷（Booklet Printing）803に変更するものであった場合について説明する。このとき、内部構造体305のメンバとして存在するCollate、Group、Staple、Layoutの各メンバのコンフリクト処理ルール適用前、すなわち、設定変更要求前における値は、以下のようになっていた。

【0054】

Collate OFF

Group ON

Staple OFF

Layout 1-Sided

【0055】

ユーザの変更要求がLayoutを1-SidedからBookletに変更するものであるので、Layoutのメンバの内容が変更されて、内部構造体305の各メンバの値は次のようになる。

【0056】

Collate OFF

Group ON

Staple OFF

Layout Booklet

【0057】

すると、プリンタドライバUI306はコンフリクトマネージャ303をコールし、状態変数リスト304にあるLayoutの状態変数が更新され、続いて推論エンジン302がコールされて、コンフリクト処理規則の適用が始まる。まず、推論エンジン302内の各プリンタ機能名が状態変数リストの各メンバの持つ値で初期化される。続いて、図6の(5)が適用され、StapleはOFFのままコントロールはdisable、すなわちグレイアウト表示されるとともに設定不可の状態にされる。同様に、同図の(7)、(8)が適用され、Collateの値はOFFからONへ、Groupの値はONからOFFへ変更された後に、各コントロールはdisableとなる。

Collate ON (disable)

Group OFF (disable)

Staple OFF (disable)

Layout BOOKLET

【0058】

以上で推論エンジン302でのコンフリクト処理規則の適用が終了する。

【0059】

コンフリクトマネージャ303は、上記ステップS507のコンフリクト処理

ルールを適用結果に基づき状態変数リストを更新し（ステップ S 5 0 8）、内部構造体 3 0 5 を更新する（ステップ S 5 0 9）。更新された箇所はプリンタドライバ UI 3 0 6 に通知される（ステップ S 5 1 0）。通知の方法としては、例えば、内部構造体 3 0 5 に更新を示すフラグを立てる、更新を示すビットフラグまたは構造体を別途用意して返す、変更箇所を示す識別子のリストを返す、プリンタドライバ UI 3 0 6 からの問い合わせに対して応答する、などいくつかの方法が考えられる。

【 0 0 6 0 】

続いて、プリンタドライバ UI 3 0 6 が内部構造体 3 0 5 のメンバの値を参照して、UI の更新が必要かどうかの判別を行う（ステップ S 5 1 1）。UI の更新の必要がない場合には、ステップ S 5 0 5 に戻り、処理を繰り返す。UI の更新が必要な場合には、UI の更新を行ったうえで（ステップ S 5 1 2）、ステップ S 5 0 5 に戻り、処理を繰り返すことになる。上記の例では、Layout が 1 - Sided Printing から Booklet Printing に設定が変更されたことにより、Collate が OFF から ON へ、Group が ON から OFF へと変化するとともに Collate, Group, Staple が disable にされるので、印刷設定画面は、図 7 に示した状態から図 8 に示す状態に更新される。

【 0 0 6 1 】

ここで、UI の更新はコンフリクトマネージャ 3 0 3 が行ってもよいし、メインプログラムであるプリンタドライバ UI 3 0 6 が行ってもよい。そのため、ステップ S 5 1 0 の変更箇所の通知はステップ S 5 1 2 の UI 更新後になる場合もある。プリンタドライバ UI 3 0 6 が変更箇所情報を知ること、プリンタドライバ UI 3 0 6 で更新処理を行う場合には、該当するコントロールのみに絞った更新処理を行うことが可能になる。また、コンフリクトマネージャ 3 0 3 で更新処理を行う場合には、プリンタドライバ UI 3 0 6 は、当該箇所の変更に伴うコンフリクト処理以外の関連処理のみを専ら行うことができるため、全体の処理効率が向上することになる。

【 0 0 6 2 】

以上の処理は、印刷設定画面がクローズされるまで、繰り返し実行される。印

刷設定画面がクローズされると処理は終了し、印刷処理関連プログラム 2 0 4 の処理も終了し、RAM 2 からは OS 2 0 5 の機能により消去される。

【 0 0 6 3 】

また、上記した処理において、ステップ S 5 1 0 ないし S 5 1 2 において、プリンタドライバ UI 3 0 6 は、どのコントロールを変更し、その結果、どのコントロールが影響を受けたのかを知ることができるため、コントロールを disable にする条件を掌握することで、図 9 に示すようにコントロールが利用不可となっている理由を表示させることも可能である。

【 0 0 6 4 】

【他の実施形態】

なお、上述した実施形態では、プリンタ装置に対して、コンフリクト処理を含む UI 制御を行う例について説明したが、本発明は、プリンタ装置に限らず、デジタルカメラ、デジタルレコーダ、イメージスキャナ等の周辺装置、制御機器の他、モデムやルータといったネットワーク関連機器にも適用可能であることはいうまでもない。これらの複数の機器から構成されるシステムに適用することも可能である。

【 0 0 6 5 】

上述したように、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（または CPU や MPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成される。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム（OS）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【 0 0 6 6 】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【 0 0 6 7 】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明した図5のフローチャートに対応するプログラムコードが格納されることになる。

【 0 0 6 8 】

【発明の効果】

以上説明したように、本発明によれば、コンフリクト処理用プログラムにおけるメインプログラムとコンフリクトマネージャとの独立性を保ったまま、コンフリクト処理によって変更された項目の情報のみを両者間で受け渡し可能とすることで、メインプログラムの処理効率を向上させるユーザインタフェース制御装置および方法ならびに記憶媒体を提供することができる。

【図面の簡単な説明】

【図1】

実施形態に係る印刷処理システムのブロック構成図である。

【図2】

実施形態におけるRAM2のメモリマップを示す図である。

【図3】

実施形態におけるプリンタドライバUI制御モジュールの概略構成図である。

【図4】

実施形態におけるプリンタドライバUI制御モジュールで扱われるデータの関連を説明する模式図である。

【図5】

実施形態におけるプリンタドライバUI制御モジュールの処理を示すフローチ

ャートである。

【図 6】

実施形態におけるコンフリクト処理ルールの一例を示す図である。

【図 7】

実施形態における印刷設定画面の一例を示す図である。

【図 8】

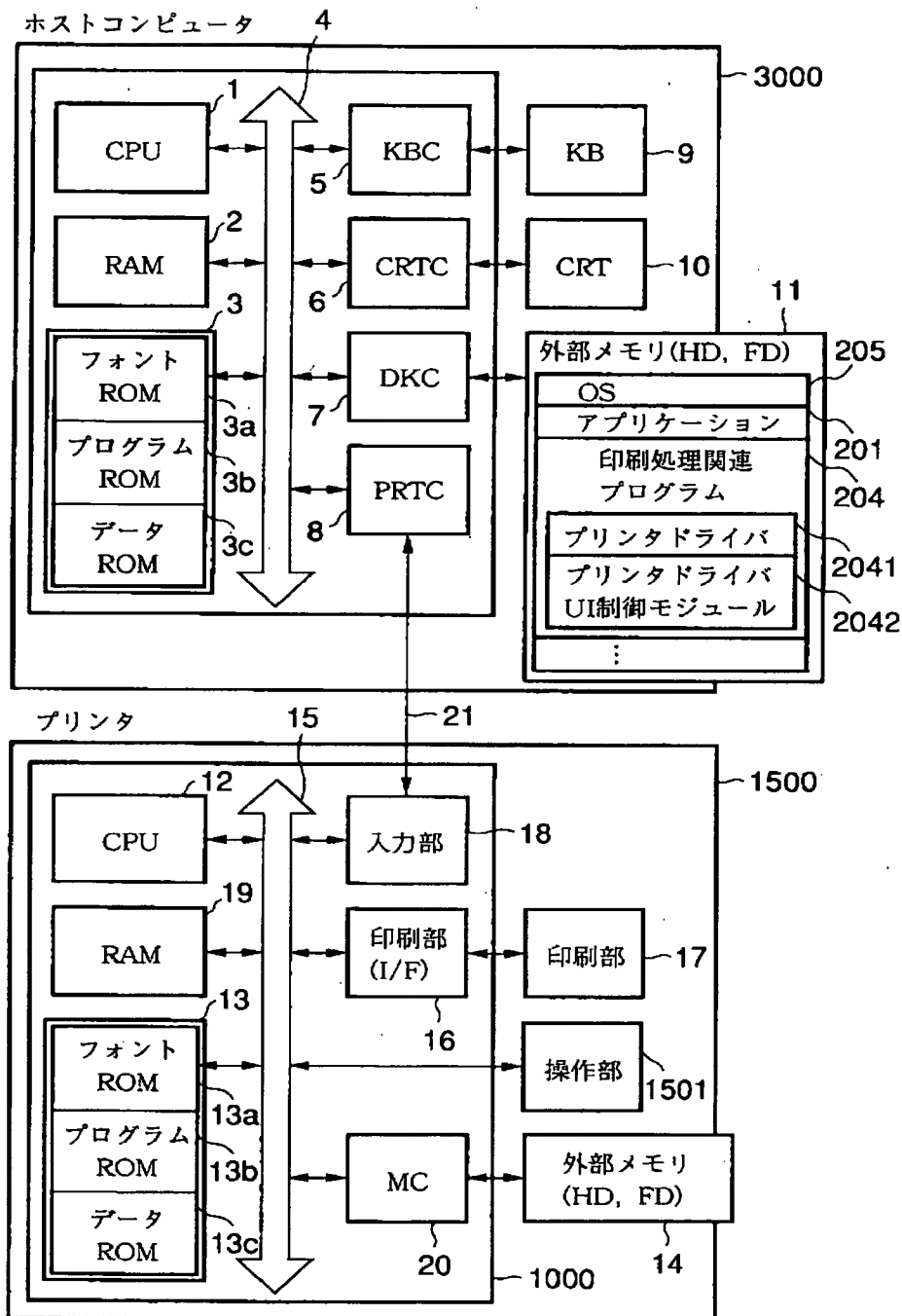
実施形態における印刷設定画面の一例を示す図である。

【図 9】

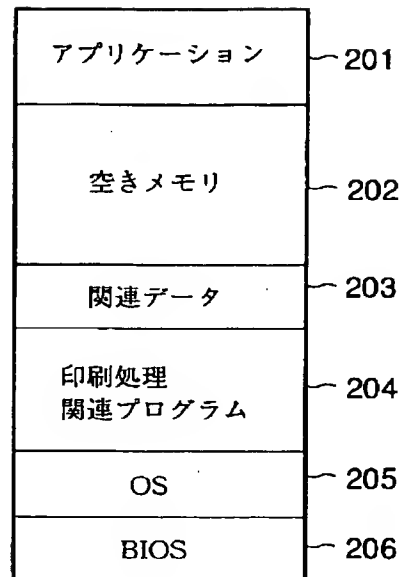
実施形態における印刷設定画面の一例を示す図である。

【書類名】 図面

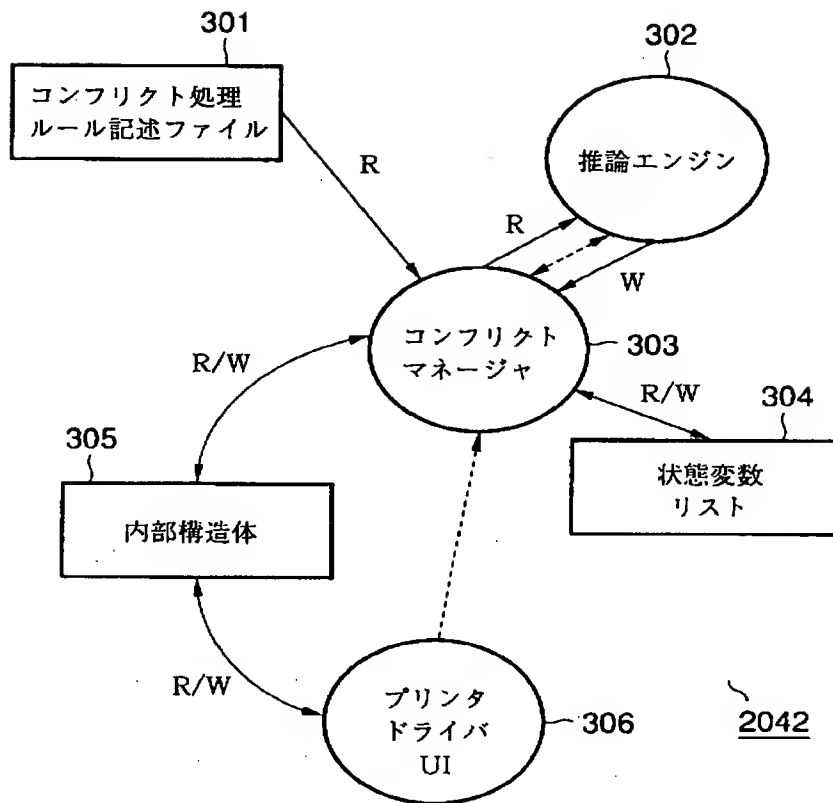
【図 1】



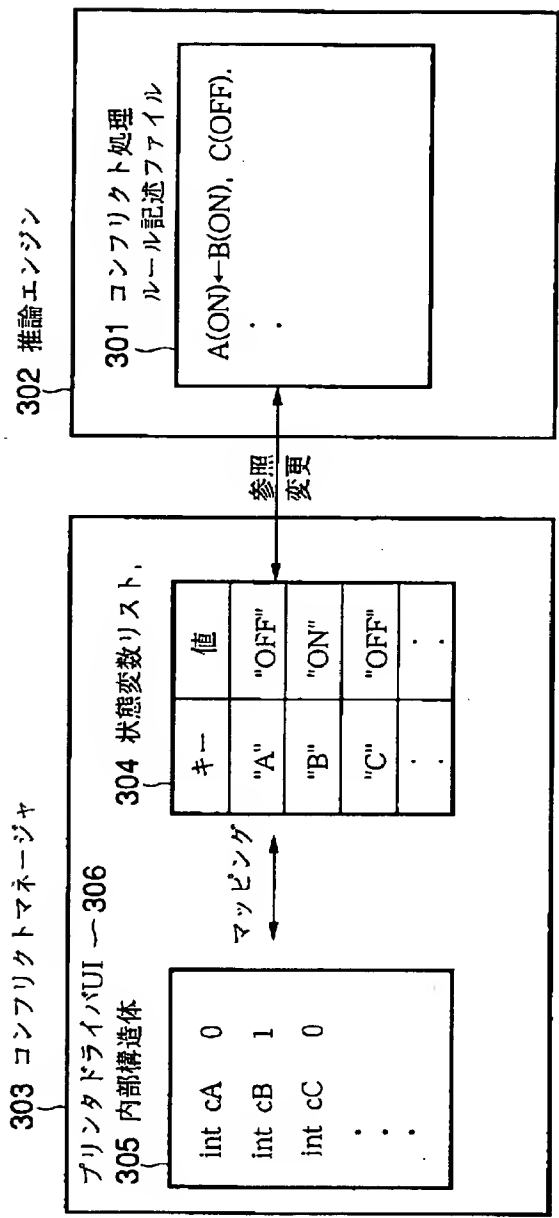
【図 2】



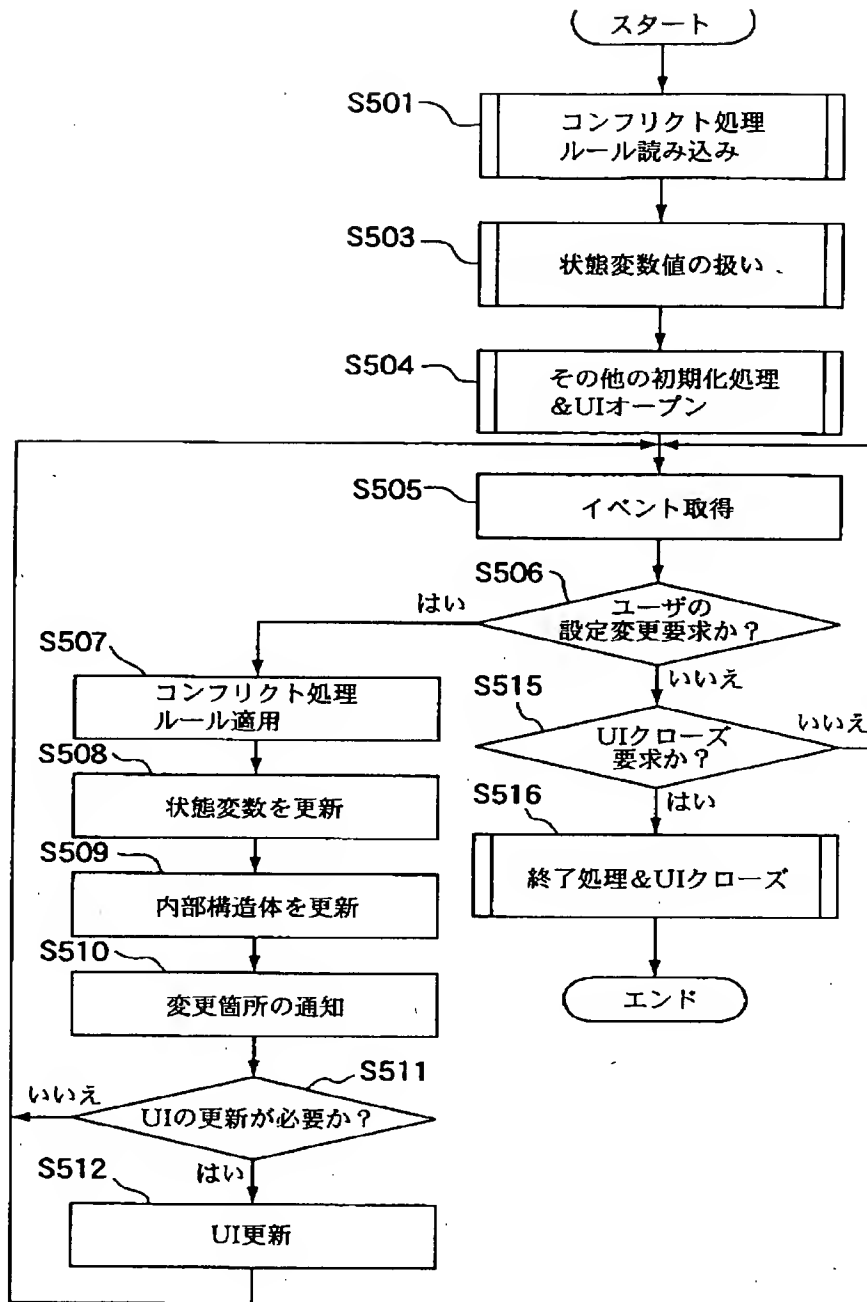
【図 3】



【図 4】



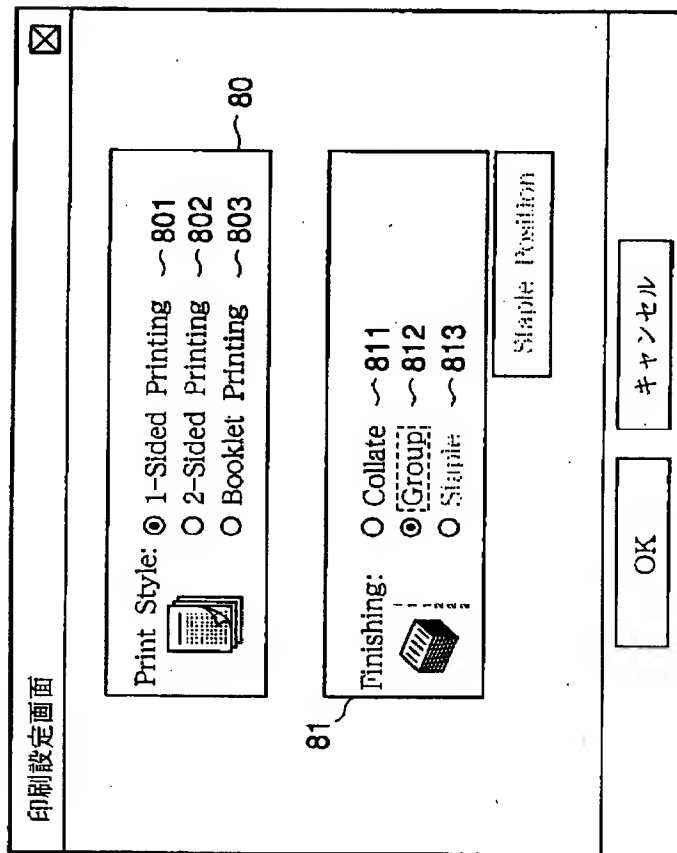
【図 5】



【図 6】

Staple(OFF)<-Collate(ON).	(1)
Staple(OFF)<-Group(ON).	(2)
Staple(OFF)<-Finisher(OFF).	(3)
{disable}	
Staple(OFF)<-StaplablePaper(OFF).	(4)
{disable}	
Staple(OFF)<-Layout(BOOKLET).	(5)
{disable}	
Staple(OFF)<-PaperType(OHP).	(6)
{disable}	
Collate(ON)<-Layout(BOOKLET).	(7)
{disable}	
Group(OFF)<-Layout(BOOKLET).	(8)
{disable}	

【図 7】



【図8】

印刷設定画面

Print Style:

1-Sided Printing

801

2-Sided Printing

802

Booklet Printing

803

Finishing:

Collate

81

Group

Staple

Staple Pillion

OK

キャンセル

【図9】

印刷設定画面

Print Style :

801

1-Sided Printing

802

2-Sided Printing

803

Booklet Printing

Finishing :

Collate

Group

Staple

OK

キャンセル

Booklet Printingが設定されているため変更できません

【書類名】 要約書

【要約】

【課題】 コンフリクト処理用プログラムにおけるメインプログラムとコンフリクトマネージャとの独立性を保ったまま、コンフリクト処理によって変更された項目の情報のみを両者間で受け渡し可能とすることで、メインプログラムの処理効率を向上させるユーザインタフェース制御装置および方法ならびに記憶媒体を提供する。

【解決手段】 設定データの中に生じる不整合を回避するための戦略を示すコンフリクト処理ルールがコンフリクト処理ルール記述ファイル（301）に記述される。推論エンジン（302）はこのコンフリクト処理ルール記述ファイル（301）を参照し、入力設定値がいずれかのコンフリクト処理ルールの適用対象となったときに当該ルールを適用する。これに応じてユーザインタフェース（UI）が更新される。UIの更新はコンフリクトマネージャ（303）が行ってもよいし、プリンタドライバUI（306）が行ってもよい。プリンタドライバUI（306）で更新処理を行う場合には、プリンタドライバUI（306）が変更箇所情報を知ることによって該当するコントロールのみに絞った更新処理を行うことができる。また、コンフリクトマネージャ（303）で更新処理を行う場合には、プリンタドライバUI（306）は、当該箇所の変更に伴うコンフリクト処理以外の関連処理のみを専ら行うことができるため、全体の処理効率が向上することになる。

【選択図】 図3

特2000-402713

出 願 人 履 歴 情 報

識別番号

[000001007]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都大田区下丸子3丁目30番2号
氏 名 キヤノン株式会社

BEST AVAILABLE COPY